MASTER OF SCIENCE IN

FINANCE

MASTER'S FINAL WORK

DISSERTATION

Pricing options using the XGBoost Model

JOÃO DIOGO MARQUES FERRAZ

SEPTEMBER - 2022

MASTER OF SCIENCE IN

FINANCE

MASTER'S FINAL WORK

DISSERTATION

Pricing options using the XGBoost Model

JOÃO DIOGO MARQUES FERRAZ

SUPERVISION:

PROF. JOÃO AFONSO BASTOS

SEPTEMBER - 2022

I am very thankful to my parents
for investing in my education and
for motivating me throughout
my entire learning journey.
Thereby, I dedicate my thesis to them.

GLOSSARY

ETF - Exchange-Traded Fund

ISIN - International Securities Identification Number

XGBoost - Extreme Gradient Boost

BSM - Black-Scholes Model

ML - Machine Learning

LGBM - Light Gradient Boosting Machine

NN - Neural Network

RMSE - Root Mean Square Error

NMSE - Normalized Mean Square Error

MAE - Mean absolute error

SVR - Support Vector Machine

DOTM - Deep Out-of-Money

OTM - Out-of-Money

ATM - At-the-Money

ITM - In-the-Money

DITM - Deep In-the-Money

ABSTRACT

Options are financial derivatives used for risk management and speculation, for example, and have been studied extensively in order to forecast its price. Before the technological revolution, parametric models were used with strict assumptions to forecast the options price, such as the Black-Scholes Model. Since then, Machine Learning models, such as the XGBoost model have been created to make forecasts without such strict assumptions from parametric models.

The purpose of this dissertation is to show how the XGBoost model can forecast option prices accurately using variables from the BSM. In addition, by using the structure of the standard deviation, third and fourth moment of the distribution of the stock price, the days to the next dividend and the next dividend to be paid, this study aims to understand by how much it improves the price forecasted from the XGBoost model with variables from the BSM. Thus, options from 100 of the biggest companies in the S&P 500 between November and February of 2020 are used to train and test the two XGBoost models. The BSM is used as the benchmark.

The results are very favorable towards the XGBoost models since the RMSE of the first and the second model are lower than the BSM by 29.51% and 35.47% , respectively. When looking at the options by its distance to the strike price, the XGBoost models always perform better than the BSM, but when the latter has a terrible performance for ITM, XGBoost has a bad performance too. For OTM put options, BSM underprices the options while XGBoost models don't. For short maturities, the XGBoost models don't improve the performance relative to the BSM by much. Although, they provide a good forecast when compared to BSM for options with long maturities. In a nutshell, the second XGBoost model is always better than the first and almost always they forecast with better accuracy and less bias than the BSM.

KEYWORDS: Options; Black-Scholes model; XGBoost; forecast

JEL Codes: G1; G12; G17; G23

Resumo

Opções são derivativos financeiros usados para gestão de risco e especulação, por exemplo, e têm sido estudadas extensivamente de modo a realizar a previsão do seu preço. Antes da revolução tecnológica, os modelos paramétricos eram usados com pressupostos restritivos para prever o preço das opções, tais como o Modelo de Black-Scholes. Desde então, os modelos de Machine Learning, tais como o Modelo XGBoost têm sido criados para realizar previsões sem pressupostos tão restritos como os modelos paramétricos.

O objetivo desta dissertação é mostrar como o modelo XGBoost pode prever o preço das opções com precisão usando variáveis do BSM. Além disso, ao usar a estrutura temporal do desvio padrão, terceiro e quarto momento da distribuição do preço da ação, os dias até ao próximo dividendo e o próxumo dividendo a ser pago, este estudo tem como objetivo perceber o quão melhor é a previsão do modelo XGBoost com as variáveis do BSM. Assim, as opções das 100 maiores empresas no S&P 500 entre Novembro e Fevereiro de 2020 são usadas para treinar e testar os dois modelos XGBoost. O BSM é usado como benchmark.

Os resultados são muito favoráveis aos modelos XGBoost uma vez que o RMSE do primeiro e segundo modelo são mais baixos do que o BSM em 29.51% e 35.47%, respectivamente. Quando se olha para as opções em termos de distância ao preço de exercício, os modelos XGBoost têm sempre uma previsão melhor que o BSM, mas quando o último tem uma péssima performance para ITM, o XGBoost tem uma má performance também. Para opções put OTM, o preço do BSM é, em média, inferior ao preço justo das opções, enquanto que os modelos XGBoost não. Para opções de curta duração, os modelos XGBoost não melhoram a performance relativamente ao BSM por muito. Contudo, os modelos dão uma boa previsão quando comparado com o BSM para opções com maturidades maiores. Em suma, o segundo modelo XGBoost é sempre melhor que o primeiro e quase sempre prevêem com melhor exatidão e menos enviesamento que o BSM.

KEYWORDS: Opções; Modelo Black-Scholes; XGBoost; Previsão

Códigos JEL: G1; G12; G17; G23

Contents

List of Figures

List of Tables

Acknowledgments

Despite this thesis being personal, other people and institutions helped me a lot directly and indirectly. Therefore, I am very thankful to:

My parents, Paulo and Maria, for investing in my education and incentivizing me through my entire learning journey since the years.

My girlfriend, Ana Gonçalves, for her presence and incentive during the writing process of the dissertation, but also for her understanding that a lot of time we could have spent together had to be used to write. This time will definitely be compensated in the future.

My supervisor, João Afonso Bastos, for helping me during the entire process of developing the models and writing, but also for his patience analyzing the data and reviewing my paper.

The teachers of ISEG for their valuable lessons and their time spent helping me throughout the challenges of the last five years.

Raquel Gaspar who gave me options data from her paper (Gaspar et al., 2020) that helped me start doing some experiments with the XGBoost model.

My friend João Vieria for recommending me my supervisor and helping me dealing with the Latex software, but also Miguel Prata for incentivizing me doing the thesis together at the university library.

All my friends from the Bachelor and the Masters for the amazing moments shared together.

1 Introduction

In the financial world, financial derivatives are widely used as tool for managing risk, speculate and create financial products. At the corporate level, these are used to attract investors, with warrants, provide incentives to employees, with stock option plans, but also for managing business risks, among other uses. Put and call options are amongst the multiple financial derivatives used.

An option is a financial derivative that gives the buyer the right to either buy or sell an underlying asset, upon the payment of a premium. A call option gives the buyer the right to buy an asset and a put option gives the buyer the right to sell an asset. On the other hand, the seller of the option has the obligation to take the opposite side until the maturity date at a price previously agreed, which is the strike price. In this study, equity options will be considered, which splits in two types: European type and American type. The buyer of an European option can only exercise its right at the maturity date, but the buyer of an American option can exercise its right at any time. In this study the options used are American type.

Over the years, academics have been developing models to estimate the fair price of option prices, but the first and most famous one is the Black-Scholes Model (Black and Scholes, 1973). It is well recognized and widely used for pricing options and its greeks. Based on the formulas of the model, the variables that affect option prices are the Stock Price (S), the Strike (K), the dividend yield (q), the risk-free interest rate (r), the stock volatility (σ) and the time to maturity (t). The greeks are measurements of sensibility to external variables, such as time, interest rates, volatility and the price of the underlying asset, that use the assumptions of the BSM. These are very useful when using options to manage risk for example.

The aim of this study is to create a model to estimate the fair value of put and call options better than the Black-Scholes model (from now onward BSM). It compares the accuracy of the BSM with the XGBoost model using the same variables, but also introduces a second XGBoost model with more variables. The BSM uses a set of mathematical formulas to compute the final price with the inputs, in order to estimate the fair value, based on a set of assumptions. The BSM formulas can be used to estimate the price of European Options, but the binomial tree in section 4.2 is used for American options, according to Hull (2014, p. 374). On the other hand, the XGBoost model uses data from other option contracts to create binary rules in the form of trees.

As explained in chapter 2, after the development of the Black-Scholes model, other researchers have tried to create models with less assumptions by avoiding the fallacy of having constant variables. Although, these models are computationally costly, need multiple optimization processes and still have some assumptions (Ivașcu, 2021).

As the technology for processing models evolves, the field of Artificial Intelligence has been evolving as well. These data driven models are computationally costly as well but have no assumptions.

This dissertation is organized in 5 chapters. In chapter 2, the development of the XGBoost model and the origin of BSM are explained as well as the limitations and assumptions of the latter. In chapter 3 the variables and the filters used are explained, the characteristics of the train and test data are analyzed and compared. In chapter 4, the way XGBoost models and BSM work is explained as well as why there are modifications to the original BSM. It also covers the XGBoost parameters. Finally, chapter 5 and 6 contain graphs that help understanding the forecasts for each model under different circumstances and for the XGBoost model the breakdown of the impact of each variable.

2 Literature Review

2.1 An Overview of Parametric Models

Before exploring other research papers related to the Black-Scholes, it is important to understand what the assumptions are in order to understand how realistic they are. Its main assumption is: “If options are correctly priced in the market, it should not be possible to make sure profits by creating portfolios of long and short positions in options and their underlying stocks.” (Black and Scholes, 1973). Also, ideal conditions in the market are assumed: short-term interest rates are known and constant through time, the price of the underlying asset follows a *random walk*, the variance of the rate of change is constant, no dividends are distributed to shareholders and the option has to be European type. Also, the model ignores transactions on the purchase of either stocks and options and ignores any penalty to sell short a stock. Under the Black-Scholes assumptions it is possible to borrow money to purchase the underlying asset but at the cost of the short term rate. It is not hard to understand that some of these assumptions are very restrictive and unrealistic, which is why other models that have weaker assumptions were created.

Merton (1973) changed the original formula and its restrictions to consider the payment of dividends and changes in strike prices. Corrado and Su (1996) adjust the Black-Scholes formula for European options with skewness and kurtosis adjustment terms as well as Jarrow and Rudd (1982) who use a Edgeworth expansion for American options. Hull and White (1987) price European call options on a stock with a stochastic volatility. Derman and Kani (1994) consider the stock price return distribution follows a binomial tree according to the volatility smile and Naik (1993) considers random jumps in volatility to accommodate for changes in macroeconomic data. Bakshi et al. (1997) show that an important factor is to use stochastic volatility, with stochastic interest rate processes having the best performance for longer maturities and stochastic random jump processes having the best performance for options with shorter maturities.

2.2 An Overview of Decision Tree Models

The XGBoost model is a Machine Learning (from now onward ML) model that uses decision trees. The model produces a forecast by using binary rules in the form of a tree. For each tree the error is computed, which will be used sequentially in the next trees. Besides the XGBoost model, there are many other tree based models such as random forests, Light Gradient Boosting Machine (as known as LGBM from now on), AdaBoost that have a similar construction and logic to the XGBoost.

Even though, decision trees models already existed, Tim Kam Ho, who worked for AT&T Bell Laboratories (Ho, 1995), set the stage for the development of the random decision forest. In his paper, he recognizes that the trees shouldn't grow too much and become too complex because it will then overfit the training data. His contribution is relevant because the models described next would overfit if there was no methods to stop it.

Few years later, AdaBoost (Freund and Schapire, 1997) was introduced with a different concept relative to other decision tree models. The model creates and combines multiple trees made of one node. In a Random Forest each decision tree is independent, thus the order doesn't matter, but for the Adabost model the order matters because each node's error influences how the next node is built. Also each tree's weight is different. Adaboost has few hyperparameters and rarely overfits if there is low noise in the data (Rätsch et al., 2001).

The Random Forest model (Breiman, 2001) uses a very distinct logic in relation to the XGBoost. The dataset is split in groups: the features are bootstrapped and the inputs are randomly sampled. This means that not all data and its respective features are in each and every single group. A decision tree is created for each group and the output is aggregated. For classification problems the result is the mode of the outputs and for regression problems the result is the average of the outputs. The model is very successful nowadays because of its simplicity, the accuracy is similar (or better) than AdaBost, the model is robust to outliers and it also provides the user with internal statistics.

A more recent model is the LGBM (Ke et al., 2017). It is the light version of XGBoost and shares similarities with it but the way the trees grow differently. While the XGBoost model grows the trees level-wise, the LGBM grows the trees leaf-wise. This makes it much faster, but still accurate.

In 2016, the XGBoost model (Chen and Guestrin, 2016) was created and is one of the most advanced decision tree models. As in the random forest model, it is possible to choose only a subset of features. Similar to the AdaBoost model, it uses boosting, meaning that the trees are created sequentially and thus the error improves after each tree. On top of boosting, XGBoost uses a gradient, which explains the name Extreme Gradient Boost. As explained in section 4.3.4, the model optimizes a loss function in order to minimize the error. Finally, the reason why XGBoost stands out is because it uses other strategies such as tree pruning, regularization, among others to drastically reduce the training time of the model and avoid overfitting, while maintaining the prediction power.

Usually ML models have a better performance than parametric models such as Black Scholes and other models previously described. When looking at the comparison from Ivașcu (2021) for options on crude oil futures, the results reported show that the Black-Scholes and Corrado-Su have the highest deviation from the price, while the ML models are more accurate and robust to outliers. In a similar paper, Park et al. (2014) compare three Parametric models: the Black-Scholes, the Heston and Merton model with two ML models: NN and Support Vector Machine (from now onward SVR) on KOSPI 200 Index options. The paper proves that the ML models don't overfit and outperform the BSM. The parametric models with weaker assumptions relative to BSM have only a slight edge when it comes to performance.

Ivașcu (2021) also compares ML models amongst each other. Decision tree models, such as XGBoost, LGBM and Random Forest have the lowest and most stable error compared to SVR and NN. For a typical model, whose input variables are the strike price, stock price, time to maturity, risk-free rate and the 60-day volatility, the Decision Tree models have a RMSE of 0.31, 0.31 and 0.39, while for NN is 0.54 and 0.59 for SVR. Furthermore, when the models are grouped for DOTM, OTM, ATM, ITM, DITM options, the 3 decision tree models have a consistent RMSE across all maturities which is a significant improvement in relation to the BSM. The NN does not have consistent RMSE across all maturities for ATM, ITM and DITM options. Across 7 non-overlapping periods, the XGBoost has the most consistent RMSE.

3 Data and Methodology

Python 3.9.2. together with the Yahoo finance API [1] were used to collect options and stock data. The Bloomberg software was used only to extract the dividend yield. The

[1] https://pypi.org/project/yfinance/

organization of the excel files, data filtering and manipulation was performed mainly with python and Microsoft Excel, but also with VBA.

The python code was able to extract 1.336.772 call and put American options on 100 companies every trading day between 11th November 2020 and 12th February 2021 [2]. The companies selected were the 100 biggest companies from the S&P500 Index, based on the holdings of the ETF IVV (ISIN: US4642872000) on the 28th October 2020.

Option contracts with volume below 20 contracts, prices below 0.05 and expiration date in the day the data was downloaded (0 days to expiration) were excluded to avoid very illiquid contracts. In order to avoid the influence of any individual company in the model, there is a cap of 10.000 option contracts for all companies. Also, the top and bottom 5% of the options by moneyness level were also excluded to avoid contracts that are not used by investors and are excessively OTM or ITM. After all the filters are applied, the total sample size is 869.501 option contracts.

For each option there are multiple independent variables and one dependent variable, which is the target variable that the model has to forecast based on the independent variables. The dependent variable is the option price divided by the notional value. The Notional Value is the strike price multiplied by 100.

The independent variables used in both models are the level of moneyness (the strike price of the option divided by the stock price), the number of days until expiration of the option, the 60-day volatility, the risk-free interest rate, a dummy variable to identify whether is put or call and the dividend yield. The interest rate used was the 3-month Treasury Bill from the Federal Reserve Bank of St. Louis

Although, the second model has more variables in order to provide more information about the changes in the distribution of the stock price and information related to the dividend yield to be paid and the number of days the dividend is due, but only when the market knows a dividend will be paid, which is the declaration date.

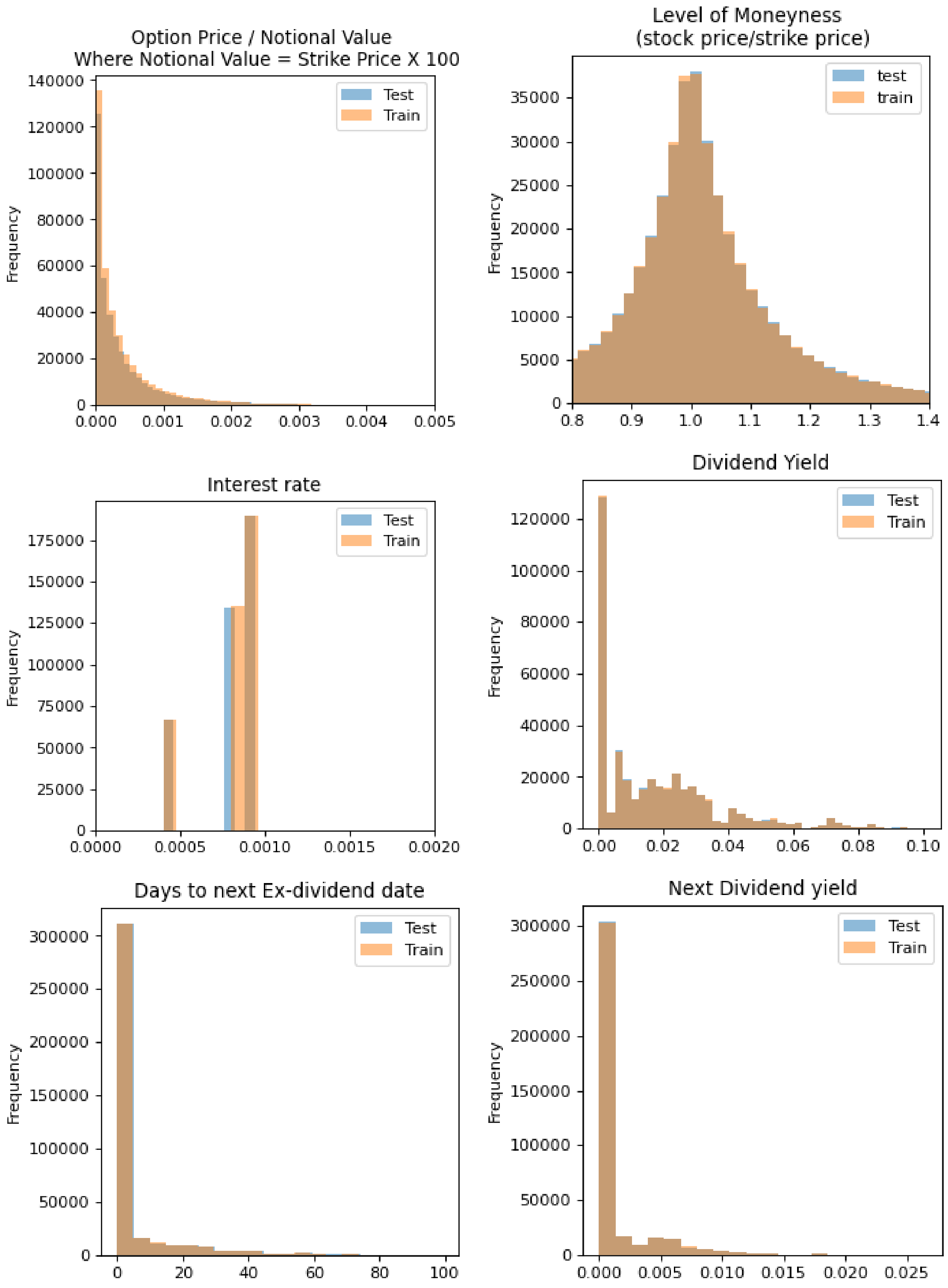
For the second model, the dividend yield isn't used, but instead the next dividend per share and the number of days until the ex-dividend date. This information is known after the declaration date, which is the date when the company announces the amount to be paid as dividend and the dates investors need to know. When a company declares the dividend it is obliged to pay it. The model 2 also includes the structure of the kurtosis, skewness and volatility of the share price. Each moment was measured for the previous 30, 60 and 90 days.

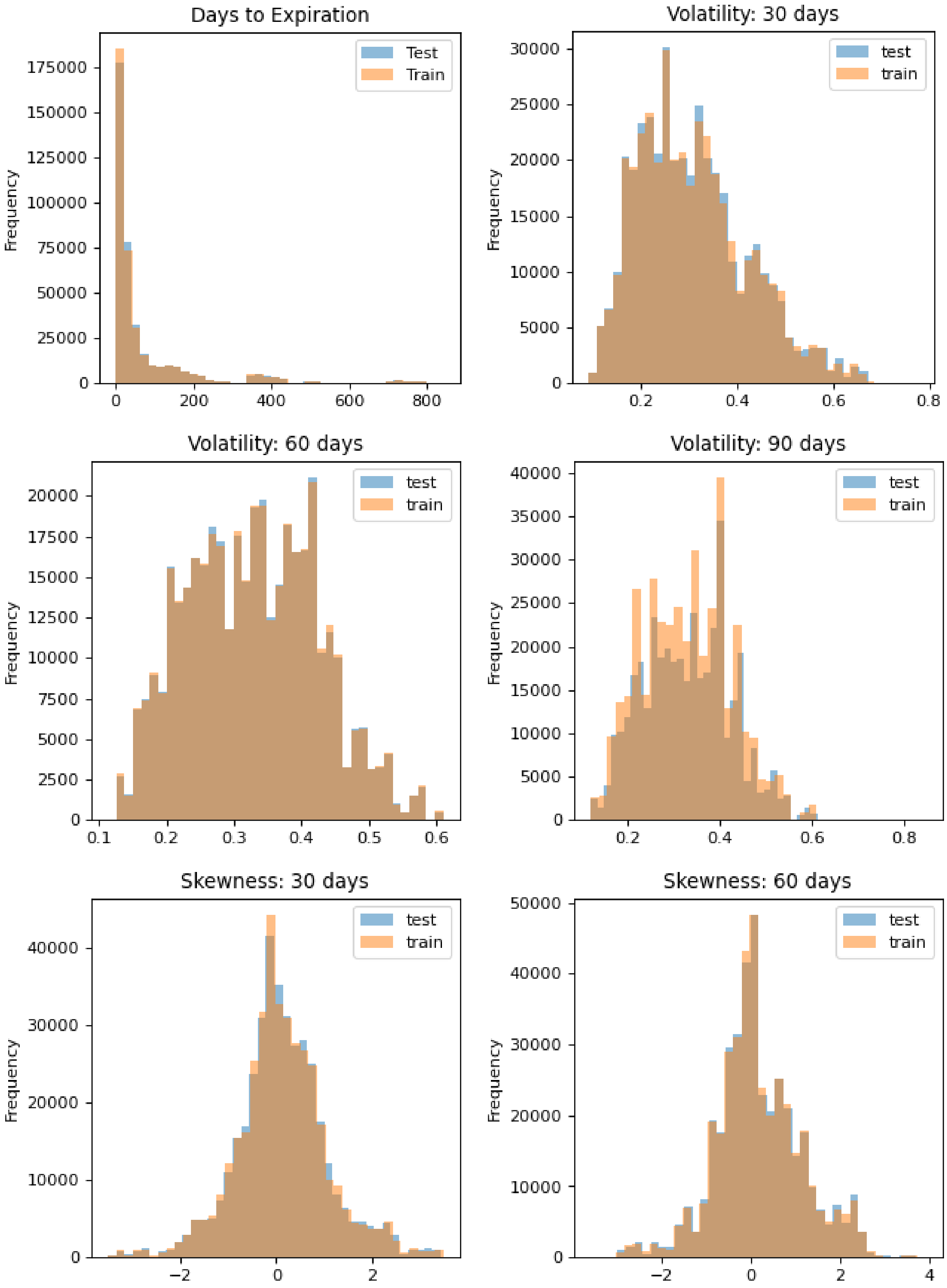
In order to train a model it is important to have the data split in training and test dataset. The training data is used to train the parameters and calibrate the model to its best version. The test dataset is used to provide an unbiased evaluation of the

[2] Except for 29th November, 7th January, 4th and 11th December

model. The data was split randomly and the train and test data have 50% each of the total sample size.

Figure 3 contains the distribution of each variable for the train and test dataset. Figure 4 contains the distribution of the train and test dataset per company.




Days to Expiration
Frequency
175000
150000
125000
100000
75000
50000
25000
0
0
200
400
600
800
Test
Train
Volatility: 30 days
Frequency
30000
25000
20000
15000
10000
5000
0
0.2
0.4
0.6
0.8
test
train
Volatility: 60 days
Frequency
20000
17500
15000
12500
10000
7500
5000
2500
0
0.1
0.2
0.3
0.4
0.5
0.6
test
train
Volatility: 90 days
Frequency
40000
35000
30000
25000
20000
15000
10000
5000
0
0.2
0.4
0.6
0.8
test
train
Skewness: 30 days
Frequency
40000
30000
20000
10000
0
−2
0
2
test
train
Skewness: 60 days
Frequency
50000
40000
30000
20000
10000
0
−2
0
2
4
test
train

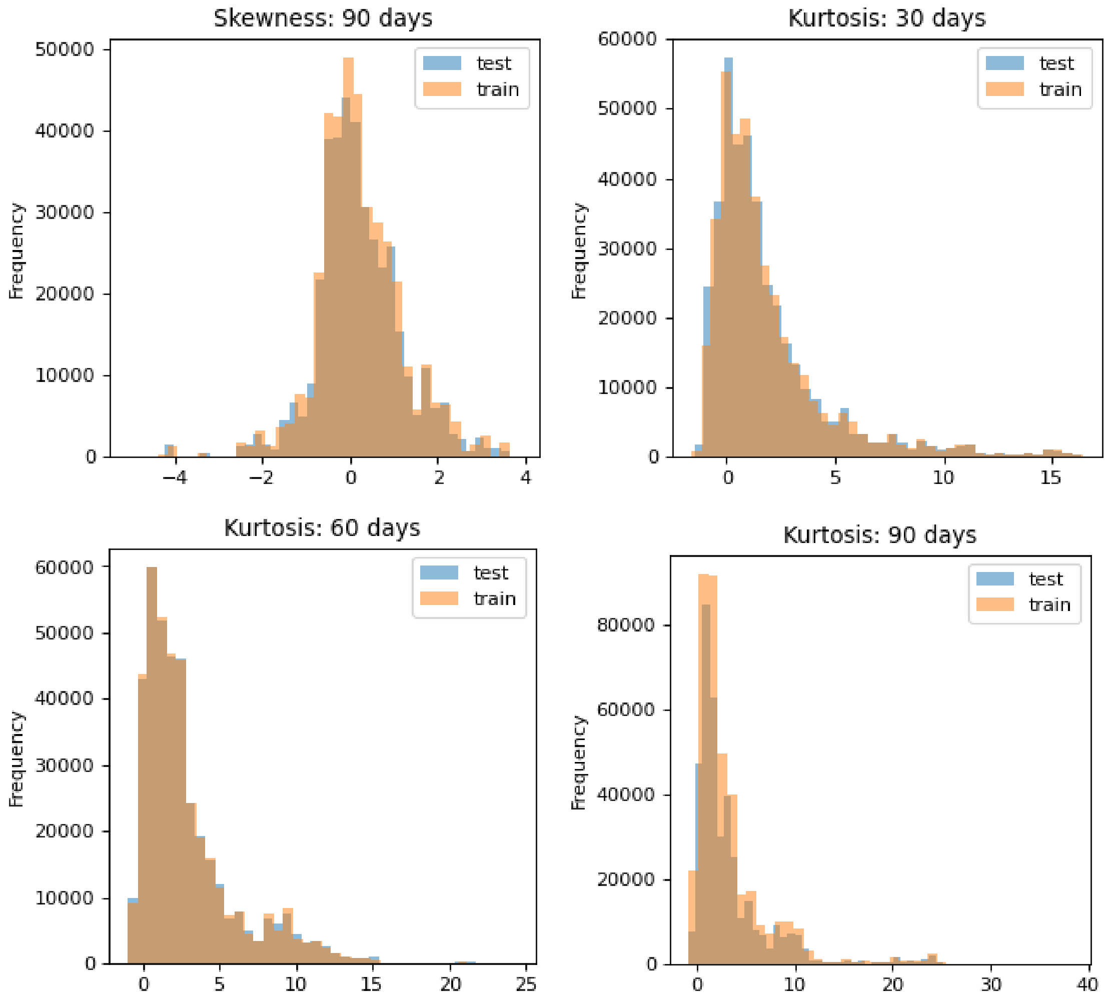


FIGURE 3: Comparision of the distribution of the test and train dataset for all variables

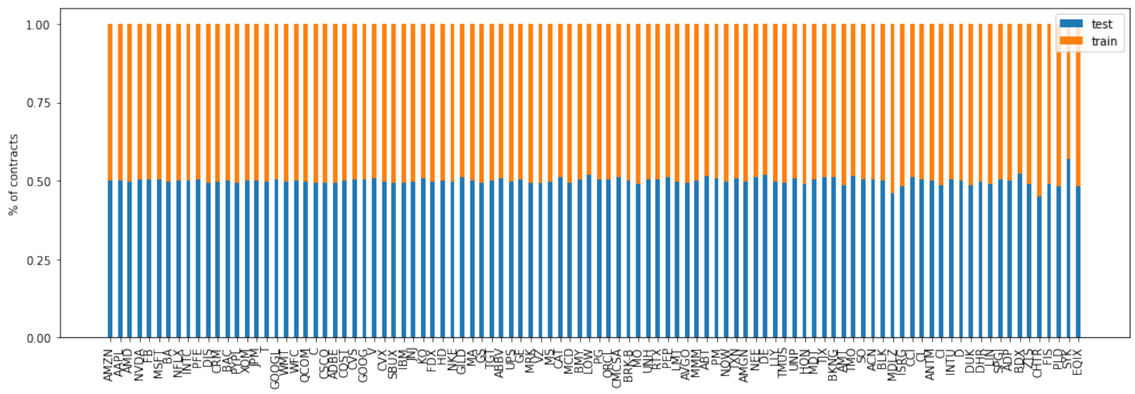


FIGURE 4: Distribution of the test and train dataset per company

By looking at the graphs in Figure 3, it is possible to see that the distribution of the features are very similar, but it's better to test for the equality of distribution in order to have an exact measure of the similarity between the training and testing dataset.

The Kolmogorov–Smirnov test was chosen to compare the distribution of the train and test dataset for each dependent variable and the independent variable. The null hypothesis for this test assumes that the two datasets have a similar distribution. Using a 95% confidence level, if the *p-value* is above 0.05, then the null hypothesis is not rejected.

According to the Table I, for all the variables the null hypothesis is not rejected, which means that all of them have similar train and test data. The random sample selection gives an assurance that the distributions of the train and test dataset are similar.

TABLE I: *p-values* of the Kolmogorov–Smirnov test for all the variables

Variables	*p-value*
Dividend Yield	0.331
Interest rate	0.9
Days to expiration	0.95
Level of Moneyness	0.778
Days to next Ex-dividend date	0.682
Next Dividend Yield	0.198
Volatility: 30, 60, 90 days	0.959, 0.204, 0.301
Skewness: 30, 60, 90 days	0.283, 0.854, 0.312
Kurtosis: 30, 60, 90 days	0.331, 0.97, 0.746
Option Price / Notional Value	0.352

4 Model Description

4.1 The Black-Scholes Model

The BSM was the first mathematical model created for pricing European options. It was developed by Fischer Black, Myron Scholes, and Robert Merton. The model assumes ideal conditions in the market that were thoroughly explained in section 2.1. These assumptions were important to help simplify the model although far from reality. The derivation of the initial equation exceeds the scope of this paper, so in order to provide a simple explanation one can tell that the calculation of an European call price is based of the following formula, according to Hull, 2014, pg. 335:

$$\mathrm{C(S,t)} = S_0\,\mathrm{N(d_1)} - \mathrm{K}\,\mathrm{e}^{-rt}\,\mathrm{N(d_2)} \tag{1a}$$

Where:

$$\mathrm{d_1} = \frac{1}{\sigma\sqrt{\mathrm{t}}}\left[\ln\left(\frac{S}{K}\right) + t\left(r + \frac{\sigma^2}{2}\right)\right] \tag{1b}$$

$$\mathrm{d_2} = \frac{1}{\sigma\sqrt{\mathrm{t}}}\left[\ln\left(\frac{S}{K}\right) + t\left(r - \frac{\sigma^2}{2}\right)\right] \tag{1c}$$

$$N(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} \mathrm{e}^{-\frac{1}{2}z^2}\,dz \tag{1d}$$

C is the price of the call option, S is the current stock price, K is the Strike price, r is the risk-free interest rate, σ is the volatility of stock returns, t is the time to maturity of the option contract in years and N is the normal cumulative distribution function.

Some of the stocks used in this dataset have dividends, thus a common practice for adjusting the Black-Scholes price for dividends is to subtract the present value of the dividend to the option price (Merton, 1973), where q is the annual dividend yield of the stock. According to Hull, 2014, pg. 373, the adjusted formula is the following:

$$\mathrm{C(S,t)} = S_0\mathrm{e}^{-qt}\,\mathrm{N(d_1)} - \mathrm{K}\,\mathrm{e}^{-rt}\,\mathrm{N(d_2)} \tag{2a}$$

Where:

$$\mathrm{d_1} = \frac{1}{\sigma\sqrt{\mathrm{t}}}\left[\ln\left(\frac{S}{K}\right) + t\left(r - q + \frac{\sigma^2}{2}\right)\right] \tag{2b}$$

$$\mathrm{d_2} = \frac{1}{\sigma\sqrt{\mathrm{t}}}\left[\ln\left(\frac{S}{K}\right) + t\left(r - q - \frac{\sigma^2}{2}\right)\right] \tag{2c}$$

$$N(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} \mathrm{e}^{-\frac{1}{2}z^2}\,dz \tag{2d}$$

The options used in this paper are not European options, but instead American ones. Thus the use of the Binomial tree is a helpful alternative to find the theorical option value, as explained in chapter 1.

4.2 The Binomial Tree Model

A Binomial Tree is a diagram that aims to represent the possible paths of the option price based on the changes of the stock price during the entire life of the option in order to compute its price at a specific point in time, assuming the stock price follows a *random walk*. With smaller time steps, the binomial tree is very close to the Black-Scholes model (Hull, 2014, p. 274) because the small time difference between each leaf subdivision makes the model similar to a continuous time model.

Figure 5 is a simplified version of the Binomial Tree, where S is the stock price and f the option price. The starting point ($t = 0$) is the beginning of the life of the option and S_0 is the stock price when the option is created. From $t = 0$ to $t = 1$, the initial node is split in two leafs: one for an higher stock price ($S_0 u$) and the other for a lower stock price ($S_0 d$), according to a predefined percentage to increase (6c) and decrease (6d). In the next time step ($t = 2$), the stock price changes by the same measure to $S_0 u^2$ or $S_0 d^2$ or $S_0 ud$. For more complex binomial trees, the nodes split the same way over time until the maturity date of the option contract.

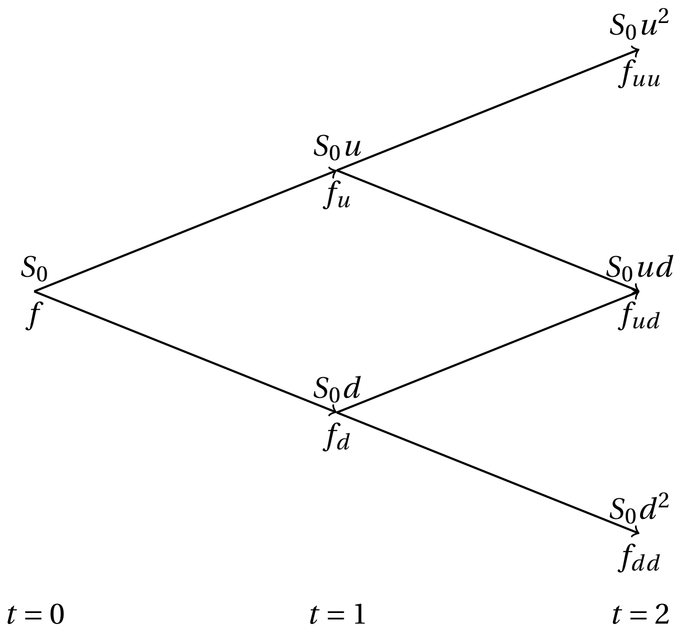


FIGURE 5: Generalized Binomial Tree with two steps

The calculation of the option price starts at the maturity date, where the price of the calls is given by $\max(S_t - K; 0)$ and the price of the puts is given by $\max(K - S_t; 0)$. Using the equation (3b), the price of the option (f) is computed backwards until $t = 0$.

Under a no-arbitrage assumption, the value of being long Δ shares and short one option (f_u), whether the stock goes up by u or down by d is given by:

$$S_0\, u\, \Delta - f_u = S_0\, d\, \Delta - f_d \tag{3a}$$

or:

$$\Delta = \frac{f_u - f_d}{S_0 u - S_0 d} \tag{3b}$$

Equation (3b) shows that Δ is the ratio between the two possible option prices and the two possible stock prices, which is the first derivative of the option price on the stock price at $t = 0$. In addition, since portfolio is riskless and under the assumption of no-arbitrage opportunities, the portfolio has to have the return of the risk free rate (r) at least. Thus the starting equation for the Binomial Tree is:

$$S_0\,\Delta - f = (S_0\,u\,\Delta - f_u)\,e^{-rT} \tag{4}$$

The goal is to find the formula for the option price at $t = 0$, but the steps taken are out of the scope of this research and it is shown in Hull, 2014, pg. 274-300:

$$f = e^{-2r\Delta t}[p^2 f_{uu} + 2p(1-p)f_{ud} + (1-p)^2 f_{dd}] \tag{5a}$$

Where:

$$p = \frac{e^{r\Delta t} - d}{u - d} \tag{5b}$$

$$u = e^{\sigma\sqrt{\Delta t}} \tag{5c}$$

$$d = e^{-\sigma\sqrt{\Delta t}} \tag{5d}$$

Considering the stock is paying a dividend at the rate of q per year, the Binomial Tree formulas have to take this effect. If the total return (dividends and capital gains) in a risk-neutral world is given by r, then the rate of the capital gains is given by $r - q$. The expected value of a stock after one period with Δt length is $S_0 e^{(r-q)\Delta t}$. This means that:

$$f = e^{-2(r-q)\Delta t}[p^2 f_{uu} + 2p(1-p)f_{ud} + (1-p)^2 f_{dd}] \tag{6a}$$

Where:

$$p = \frac{e^{(r-q)\Delta t} - d}{u - d} \tag{6b}$$

$$u = e^{\sigma\sqrt{\Delta t}} \tag{6c}$$

$$d = e^{-\sigma\sqrt{\Delta t}} \tag{6d}$$

4.3 An overview of the XGBoost model

In this section, the theory and the formulas behind the model are explained in detail, as well as how it was tuned to forecast the option prices. Section 4.3.1 covers the structure of the XGBoost model, the definitions of each component and the definitions of the parameters. Sections 4.3.2 to 4.3.5 explain how the model works theoretically. Finally, section 4.3.6 contains the practical explanation on how the three parameters in section 4.3.1 were selected and its values.

4.3.1 Model Structure and inputs

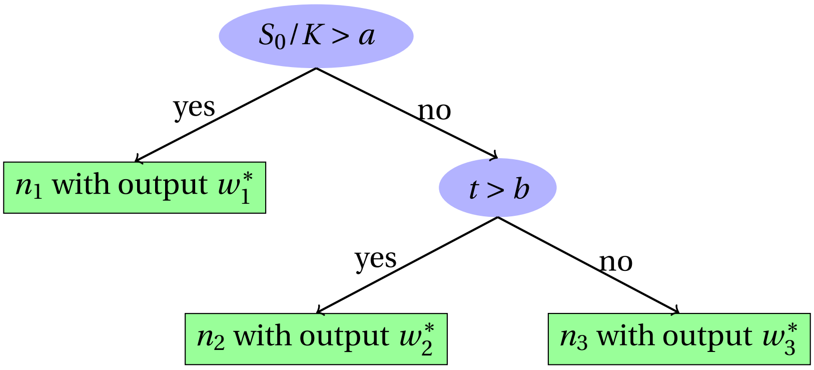

$S_0/K > a$
yes
no
n_1 with output w_1^*
$t > b$
yes
no
n_2 with output w_2^*
n_3 with output w_3^*

FIGURE 6: Simplified illustration of XGBoost Structure

Figure 6 is a simple decision tree created to provide an explanation about how the model works. It has 3 leaf nodes and 2 nodes, where $S_0/K > a$ is the root node and $t > b$ is the internal node or just node. The feature dimension (d) of the first node is S_0/K and for the second node is t. An XGBoost model is made by multiple trees, all different from each other.

There are a lot of parameters that allow the customization of the XGBoost Model. Although only three of these parameters were used while the others were set with default values:

- **eta** is the learning rate and it is used as a weighting factor for each new tree that helps prevent overfitting of the model. It is set to a range between 0 and 1.
- **max_depth** is the maximum depth of the decision tree, where each level is a group of nodes and/or leaves in the same row starting in the root node. For instance, the decision tree in Figure 6 has a depth of 2.
- **N_estimators** is the number of trees in the model, where the new tree corrects the error made by the previous trees, thus learning incrementally. So, the more trees it has the more it learns, but the odds of overfitting increase as well. The parameter has no maximum value.

4.3.2 Tree structure

A tree ensemble model (7a) with K addictive functions predicts the output of a data set (7c) with n examples and m features,

$$\hat{y}_i = \phi(X_i) = \sum_{\kappa=1}^{K} f_\kappa(X_i), f_\kappa \in \mathscr{F}, \tag{7a}$$

Where:

$$\mathscr{F} = \{f(X) = w_{q(x)}\}(q : \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T) \tag{7b}$$

$$\mathscr{D} = \{(x_i, y_i)\}(|\mathscr{D}| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}) \tag{7c}$$

$\mathscr{F}$ represents the space of the regression trees where f_κ is an independent tree structure q with T leafs and each leaf has a weight of w. Each leaf has a continuous score, thus w_i is used to represent a specific leaf. Each tree (represented by q) has decision rules to split between the leaves.

4.3.3 Regularized learning objective

In order to find the optimal weight w_j^* for each leaf, the loss function (8b) plus the regularization parameter (8c) must be minimized. Thereafter the scoring function is calculated based on the output values of the leafs and the nodes. Each split candidate is evaluated based on these scores.

$$\mathscr{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_\kappa \Omega(f_\kappa) \tag{8a}$$

Where:

$$l(\hat{y}_i, y_i) = \frac{1}{2}(y_i - \hat{y}_i)^2 \tag{8b}$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2 + \alpha|w| \tag{8c}$$

$l(\hat{y}_i, y_i)$ has to be differentiable and convex because these are the requirements to minimize the function. $\Omega(f)$ is the regularization term that penalizes the more complex the model is.

4.3.4 Gradient tree boosting and Shrinkage

The model starts with an initial prediction which is set by default to 0.5. The errors are calculated based on the difference between the observed values in the data and the predicted values, which is the average price in the node.

Thereafter the output value of the node is calculated by minimizing the Equation 8a. Firstly, the second-order Taylor Polynomial Pois used and then regularization term is expanded which results in Equation 9:

$$\tilde{\mathscr{L}}^{(t)} = \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2] + \gamma T \tag{9}$$

The function for the output value of the leafs w_j^* is given when the Equation 9 is set equal to zero, which results in Equation 10.

$$w_j^* = \begin{cases} -\dfrac{\sum_{i\in I_j} g_i + \alpha}{\sum_{i\in I_j} h_i + \lambda} & \sum_{i\in I_j} g_i < -\alpha \\ -\dfrac{\sum_{i\in I_j} g_i - \alpha}{\sum_{i\in I_j} h_i + \lambda} & \sum_{i\in I_j} g_i > \alpha \\ 0 & else. \end{cases} \tag{10}$$

The equation of the score of each leaf and node (11) is given by the substitution of the Equation 10 in the Equation 9:

$$\begin{aligned} \tilde{\mathscr{L}}^{(t)}(q) &= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)(-\frac{\sum_{i\in I_j} g_i + \alpha}{\sum_{i\in I_j} h_i + \lambda}) + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)(-\frac{\sum_{i\in I_j} g_i + \alpha}{\sum_{i\in I_j} h_i + \lambda})^2] + \gamma T \\ &= \sum_{j=1}^{T}[-(\frac{(\sum_{i\in I_j} g_i)^2 + \alpha}{\sum_{i\in I_j} h_i + \lambda}) + \frac{1}{2}\frac{(\sum_{i\in I_j} g_i)^2 + \alpha}{\sum_{i\in I_j} h_i + \lambda}] + \gamma T \\ &= -\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i\in I_j} g_i)^2 + \alpha}{\sum_{i\in I_j} h_i + \lambda} + \gamma T \end{aligned} \tag{11}$$

g_i (12a) is the gradient and h_i (12b) is the hessian, which are the first and second derivative of the loss function, respectively. In this case, $(\sum_{i\in I_j} g_i)^2$ is the sum of the residuals squared and $(\sum_{i\in I_j} h_i)$ is the number of residuals, thus the score (11) is half of the average squared residuals, but it could be different based on the value of other parameters λ, α and γ.

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) = \frac{d}{d\hat{y}_i}[\frac{1}{2}(y_i - \hat{y}_i)^2] = -(y_i - \hat{y}_i) \tag{12a}$$

$$h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) = \frac{d^2}{d\hat{y}_i^2}[\frac{1}{2}(y_i - \hat{y}_i)^2] = \frac{d}{d\hat{y}_i}[-(y_i - \hat{y}_i)] = 1 \tag{12b}$$

Given a m feature, the split candidate is evaluated by the loss reduction (13), where I_L and I_R are the left and right nodes, respectively, and $I = I_L \cup I_R$.

$$\mathscr{L}_{split} = \frac{1}{2}[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}] - \gamma \tag{13}$$

When the ideal split is found, the output (10) is used as the forecast for the price of the option.

4.3.5 Approximate Algorithm and Sparsity-aware Split Finding

Equation 13 is solved for multiple splits and features which takes a lot of time. Since there is a trade-off between accuracy and efficiency, the XGBoost creators solved this problem with an algorithm.

The exact greedy algorithm evaluates all the splits, which is computationally demanding. XGBoost supports the exact greedy algorithm, but also supports the approximate algorithm. The later proposes splitting the data based on quantiles of the feature being tested, though the observations aren't distributed evenly. Each observation has a weight equal to the hessian of the observation. Therefore, the weights in regression problems are always the same as shown in Equation 12b, unlike classification problems. In classification problems, the different weights for the observations allows the model to place the splits where there is less confidence in the forecast in order to improve it. The use of weights per observation is named as the weighted quantile sketch.

Let $\mathscr{D}_k = \{(x_{1k}, h_1), (x_{2k}, h_2)...(x_{nk}, h_n)\}$, where each value x_{nk} corresponds to a example n and a feature m and each example n has a hessian h_n. According to (Chen and Guestrin, 2016), XGBoost allows the use of the weighted quantile sketch, where the hessian is used to compute the weight of each example, which is then applied to the percentiles. Although, since the hessian of the loss function (12b) is a constant, then the weighted quantile sketch does not apply here because the weight is the same for every example.

The XGBoost model uses another algorithm named Sparsity-aware split finding. As the name implies, its goal is to make the model aware of the sparsity patterns present in the date due to frequent zero values and missing data (Chen and Guestrin, 2016). The algorithm works by setting a default direction in each tree node and the model learns what is the optimal default direction based on the data, which runs 50 times faster than the version ignoring the sparsity of data (Chen and Guestrin, 2016).

4.3.6 Hyperparameter tuning

The Hyperparameter tuning of the model is performed using the K-fold cross-validation and the number of folds is set to 10. This technique splits the data in K groups but 9 groups are used to train the model, while the remaining group is used to test the model. This process is repeated as many times as the value set for the K-fold input because each group is used as the testing model once. The scoring function was set as the negative value of the mean squared error and the average error of the 10 groups is set as the model's error.

Friedman (2001) suggests that the learning rate depends on the number of estimators. An increase in the learning rate should lead into a lower number of estimators given that the higher influence of each tree will reach the target value faster, which decreases the total number of trees needed for an accurate forecast.

Given the dependency between these two parameters they were tuned jointly and given the influence of the number of estimators in the size of the model, the maximum depth parameter was added to the tuning too, as another method to control the size.

TABLE II: XGBoost Model characteristics

	XGB Model 1	XGB Model 2
Output Variable	Option Price / Notional Value	Option Price / Notional Value
Input Variables	Price / Strike Call or Put (binary variable) DIvidend Yield Days to Expiration 60 Day Volatility Interest Rate	Price / Strike Call or Put (binary variable) Days to Expiration next dividend/share days until ex dividend date 30, 60, 90 Day Volatility 30, 60, 90 Day Skewness 30, 60, 90 Day Kurtosis
Learning Rate	0.01	0.05
Maximum Depth	15	15
Number of estimators	1600	800

There are many other parameters that don't have a big impact in the model, but add complexity to the process of choosing the best model, which makes it a more demanding process for the computer's hardware. Thus, all the other inputs were set as default.

Two of these parameters are the tree method and the objective funcion which were left with the default values, but not by randomness. According to Chen and Guestrin (2021), the tree method is set by default to "auto", which uses the approximate algorithm for large datasets. The default option was left unchanged because as it was explained in section 4.3.5 this algorithm is more efficient. As per the same source, the default value for the objective is the squared loss, which is the desired equation as explained in section 4.3.3.

5 Results and Discussion

In section 5.1, the distribution of the residuals and the errors of the three models are compared. In section 5.2, a similar analysis is performed but with more detail because it splits the data by 3 of the options characteristics. On section 5.3, the goal is to understand the impact of the variables in the forecast for both XGBoost models, using feature importance plots.

5.1 Model comparison

To compare the models, the RMSE and box and whisker plot were used in order to evaluate the expected error and its distribution. Table III presents the RMSE results. The RMSE is interpreted as the average deviation from the true option value. Overall, the second XGBoost model was the best model, while the first was slightly worse. The Black-Scholes model was worse by a wide margin than the XGBoost models.

TABLE III: MSE and RMSE (in 10^{-5}) for the three models

	XGB1	XGB2	BSM
MAE	5.8	5.3	8.2
RMSE	9.1	8.33	12.91

As shown in Figure 7, both XGBoost models have a similar distribution of errors, by looking at the first and the third quantile, with the second model having slightly tighter quantiles. The Black-Scholes model has wider quantiles and extreme values, which helps explaining the extreme value of the RMSE.

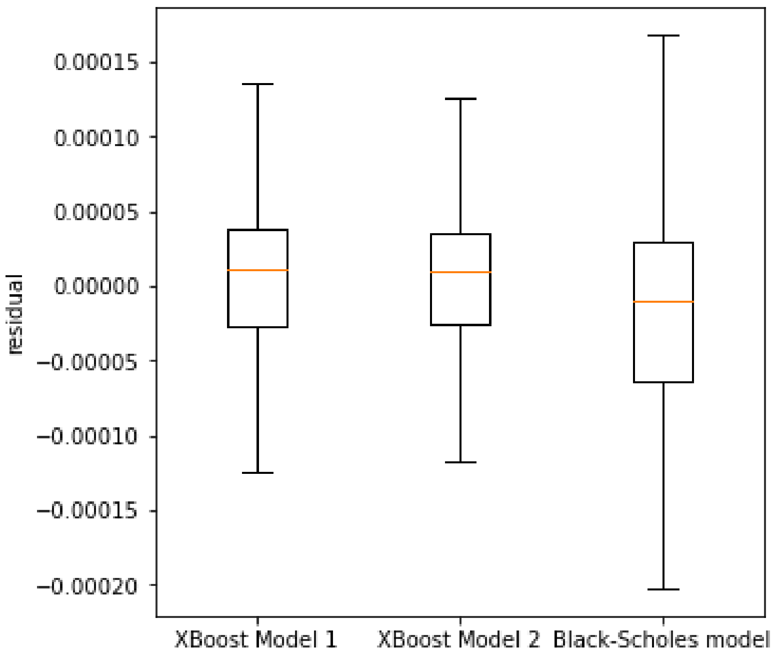


FIGURE 7: Distribution of the residuals for the XGBoost models and the Black-Scholes model

Figure 8 shows the distribution of the residuals through an histogram. It shows that the Black-Scholes model has positive skewness and high kurtosis on the left side. XGBoost models have very little skew and kurtosis, thus the distribution is more symmetric. The Black-Scholes model has a clear tendency to underprice options while the XGBoost models have little bias.

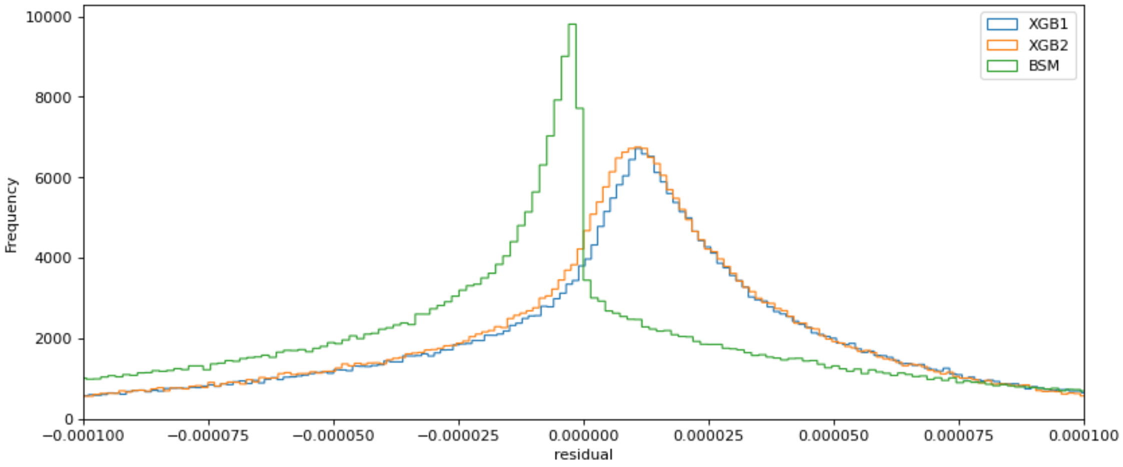


FIGURE 8: Histogram of the residuals for the XGBoost models and the Black-Scholes model

As in Culkin and Das, 2017, the forecasts of each XGBoost model and the option prices were used to compute the R^2 for train and test data to figure out if there is overfitting in the models. For both models, the values between test and train data are very similar, according to Table IV. Thus, there isn't evidence of overfitting in the XGBoost models.

TABLE IV: R-squared of the XGBoost Models with option prices

	XGB1	XGB2
Train dataset	0.9783	0.9822
Test dataset	0.9767	0.9805

5.2 Model comparison by feature

Figure 9, 10, 11 and 12 show the distribution of the residuals per quantile, in order to analyze the data with an equal distribution between samples. These quantiles are based on a set of variables, namely level of moneyness, days to expiration and Market Cap.

Figure 9 and 10 splits the errors based on the distance to the strike price for calls and puts, respectively. The ratio is defined as the stock price divided by the strike price. The first quantiles include options that have a strike price below the stock price (In-the-money for calls and out-of-money for puts) and the last quantiles include options that have a strike price above the stock price (In-the-money for puts and out-of-money for calls).

In Figure 9 the first quantile has call options with a ratio between 0.708 and 0.971, which includes all options ITM. The options in the second quantile have a ratio between 0.971 and 1.013. The options in the third quantile have a ratio between 1.013 and 1.054. The options in the fourth quantile have a ratio between 1.054 and 1.121. The fifth quantile has options whose ratio is between 1.121 and 1.325.

In Figure 10 the first quantile has put options with a ratio between 0.708 and 0.864. The options in the second quantile have a ratio between 0.864 and 0.924. The options in the third quantile have a ratio between 0.924 and 0.965. The options in the fourth quantile have a ratio between 0.965 and 1. The fifth quantile has options whose ratio is between 1 and 1.325, which includes all options ITM.

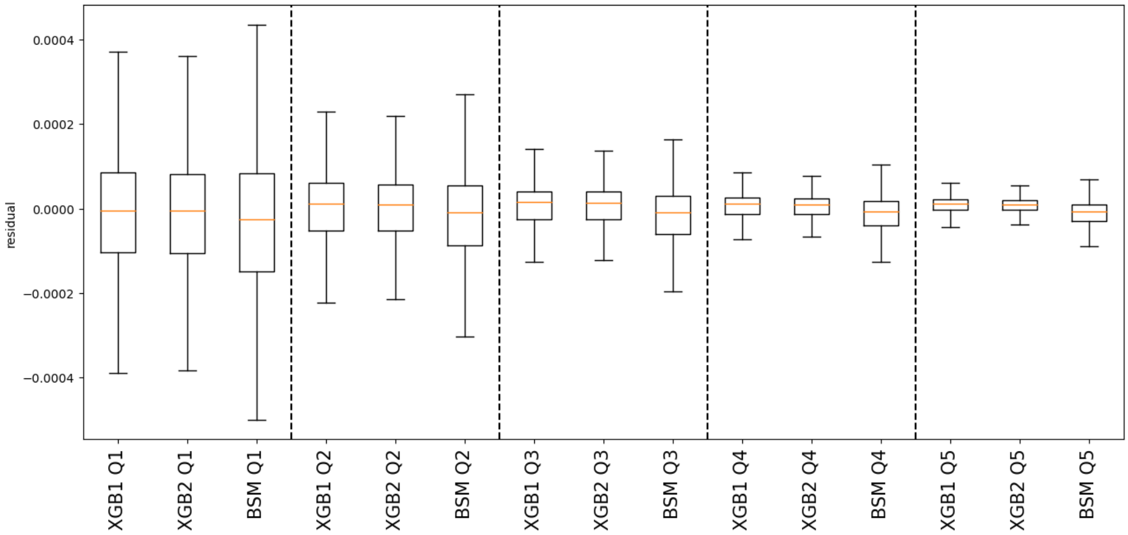


FIGURE 9: Distribution of the residuals of call options by the level of moneyness for the XGBoost models and the Black-Scholes model

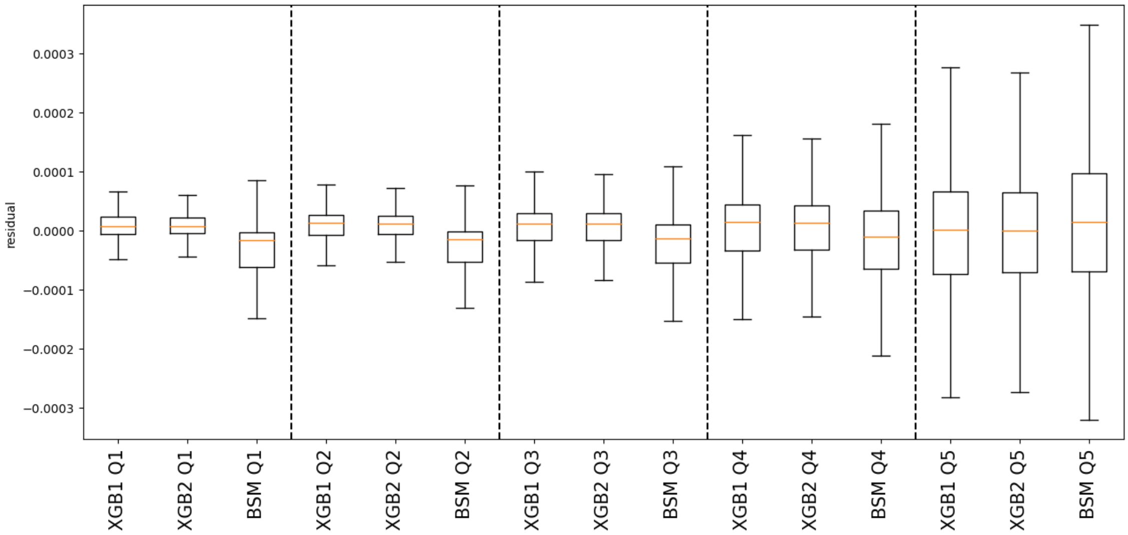


FIGURE 10: Distribution of the residuals of put options by the level of moneyness for the XGBoost models and the Black-Scholes model

For both call and put options, the accuracy of the three models worsens as the options move closer to be In-the-money. The XGBoost models are better than the Black-Scholes regardless of the quantile, according to Table V.

Although, the XGBoost models has the best performance in the first and second quantile for put options and fourth and fifth quantile for call options because the BSM will underprice OTM options, as stated in the literature (Geske and Roll, 1984). This occurrence stands out in Figure 10 because there is a bias in the distribution of put and call options. For put options the fifth quantile has options with a ratio higher than 1, while for call options the first has options with a ratio lower than 1.

According to Table V, the error of the BSM for call options improves a lot across the first four quantiles, while the error of the put options across the first four quantiles is very stable and erratic.

TABLE V: MSE and RMSE (in 10^{-5}) of the three models by level of moneyness for calls and puts

Calls	MAE					RMSE				
Quantile	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
XGB1	12.50	7.35	5.10	3.74	3.18	17.1	9.86	7.21	5.92	5.40
XGB2	11.95	6.87	4.72	3.25	2.61	16.1	9.02	6.47	4.90	4.22
BSM	15.40	9.66	7.18	5.80	5.48	20.8	13.5	11.04	9.91	9.85

Puts	MAE					RMSE				
Quantile	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
XGB1	3.16	3.46	4.06	5.53	8.93	5.43	5.41	6.10	7.60	11.91
XGB2	2.65	3.01	3.72	5.20	8.52	4.32	4.51	5.40	7.04	11.32
BSM	6.85	6.02	6.25	7.43	11.18	11.95	10.25	10.08	10.98	15.30

In Figure 11 the errors are divided in five quantiles based on the maturity of the options measured in days. The first quantile has options with an expiration date up 8 days, included. The options in the second quantile expire between 9 and 17 days. The options in the third quantile expire between 18 and 35 days. The options in the fourth quantile expire between 36 and 101 days. The fifth quantile has the widest range with options expiring between 102 and 840 days.

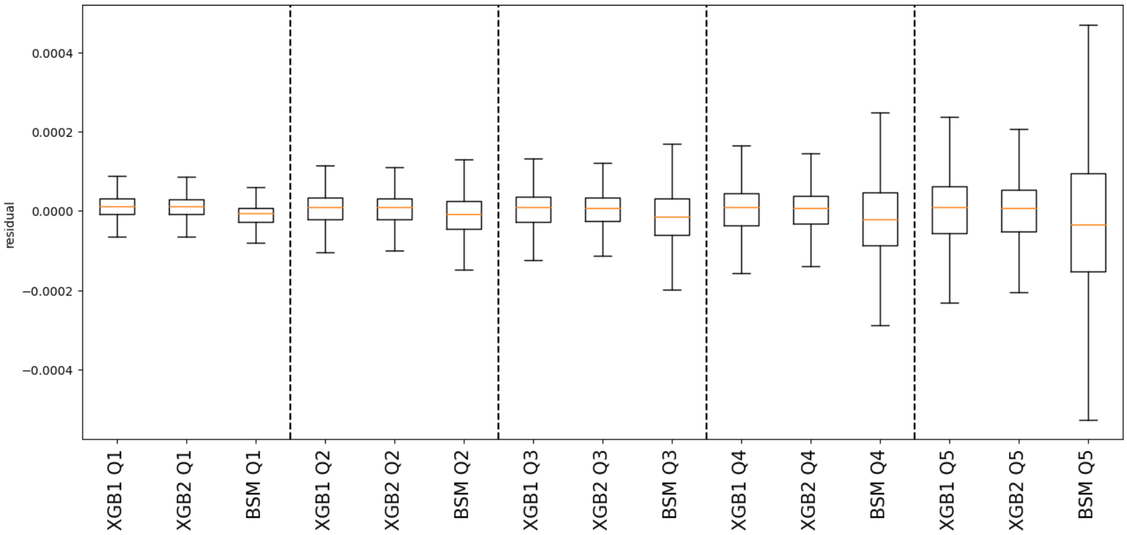


FIGURE 11: Distribution of the residuals per days to expiration for the XGBoost models and the Black-Scholes model

According to Figure 11, for the shortest maturities, the Black-Scholes model has a similar distribution of errors compared to the XGBoost models, but it usually underprices the options. As the maturity increases, the distribution of the Black-Scholes model become wider than the XGBoost models. Both XGBoost models provide are very accurate in the fourth and fifth quantiles, where Black-Scholes fails to deliver an accurate value, according to Table VI. In the first quantile the accuracy is very similar across the three models.

TABLE VI: MSE and RMSE (in 10^{-5}) of the three models by days to expiration

	MAE					RMSE				
Quantile	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
XGB1	4.92	4.94	5.33	6.26	8.47	8.24	7.86	8.20	9.30	12.12
XGB2	4.79	4.72	4.90	5.51	7.37	7.97	7.44	7.54	8.24	10.56
BSM	4.88	4.89	7.03	9.64	16.48	8.71	9.09	10.10	13.21	21.71

In Figure 12 the errors are divided in five quantiles based on the Market Cap of the underlying stock. The first quantile has companies with a Market Cap up to $91.8 Billions, included. The Market Cap range of the second quantile is between $91.8 and $150.7 Billion. The Market Cap range of the third quantile is between $150.7 and $214.8 Billion. The Market Cap range of the fourth quantile is between $214.8 and $762.5 Billion. The Market Cap range of the fifth quantile is between $762.5 and $1.9 Trillion.

Figure 12 clearly shows that all models have a similar distribution of errors, regardless of the size of the company, except for companies in the first quantile. In this first quantile the three models underprice the options. According to Table VII, the error across the 4 quantiles is nearly the same, except for the first quantile, which is slightly higher.

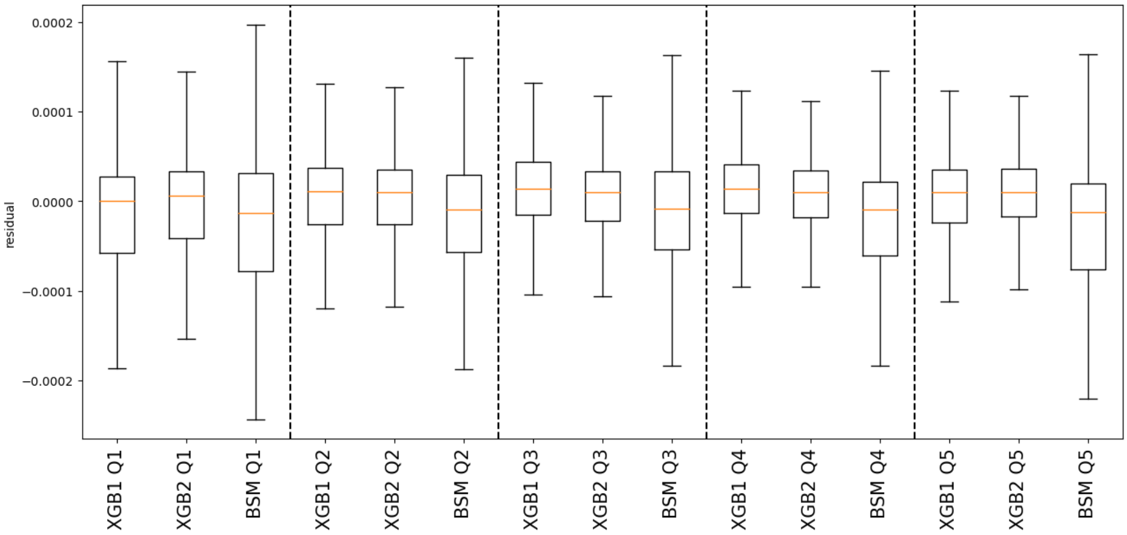


FIGURE 12: Distribution of the residuals per Market Capitalization for the XGBoost models and Black-Scholes model

TABLE VII: MSE and RMSE (in 10^{-5}) of the three models by Market Cap

	MAE					RMSE				
Quantile	Q1	Q2	Q3	Q4	Q5	Q1	Q2	Q3	Q4	Q5
XGB1	7.11	5.49	5.58	5.09	5.13	11.5	8.46	8.59	7.72	7.75
XGB2	6.23	5.33	5.06	4.65	4.84	9.87	8.25	7.91	7.16	7.36
BSM	9.78	7.64	7.74	7.21	8.25	15.44	11.95	12.07	11.26	12.64

5.3 Feature importance plots

The contribution of each variable to the model is measured through feature importance plots according to three metrics: total gain, total coverage and weight. The feature importance plots for the first XGBoost model are represented in Figure 13, 14 and 15. The feature importance plots for the second XGBoost model are represented in Figure 16, 17 and 18.

The feature importance plots represented in Figure 13 and 16 are based on the total gain, which is defined as the total gain of each feature for all the splits. This computation is then performed for all the trees of the model. The highest values imply that the feature adds the most value to the forecast. The feature importance plots represented in Figure 14 and 17 are based on the total coverage. The coverage is the relative number of samples affected by each split of feature for all trees. The feature importance plots represented in Figure 15 and 18 are based on the weight, which is the frequency the feature occurs in the splits of the trees.

According to Figure 13 and 14, the distance to the strike price and the days to expiration are 2 of the most relevant variables because both contribute a lot to improve the accuracy of the model tree after tree and both help split many samples. The dummy variable has also a big contribution to the model but since it is a binary variable it isn't used as frequently as other variables. In the Figure 15 it is the least important by weight. The same happens with interest rates that also have very few values according to Figure 3. The 60 day volatility affects many samples, is one of the most used variables but provides a very small gain, similar to the gain of the dividend yield and the interest rate. The dividend yield is also used in the model quite often despite the fact that it provides a very small gain and few samples are selected across the splits with dividend yield as a feature. Finally, the interest rate is the variable that provides the least gain, is the least used and splits the lowest amount of samples.

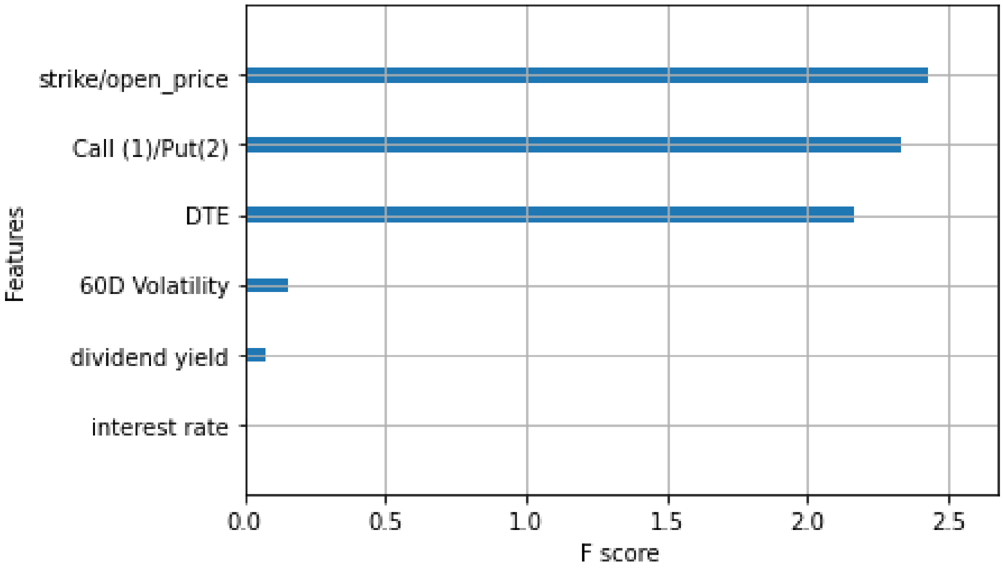


FIGURE 13: Feature importance graph by total gain for the XGBoost Model 1

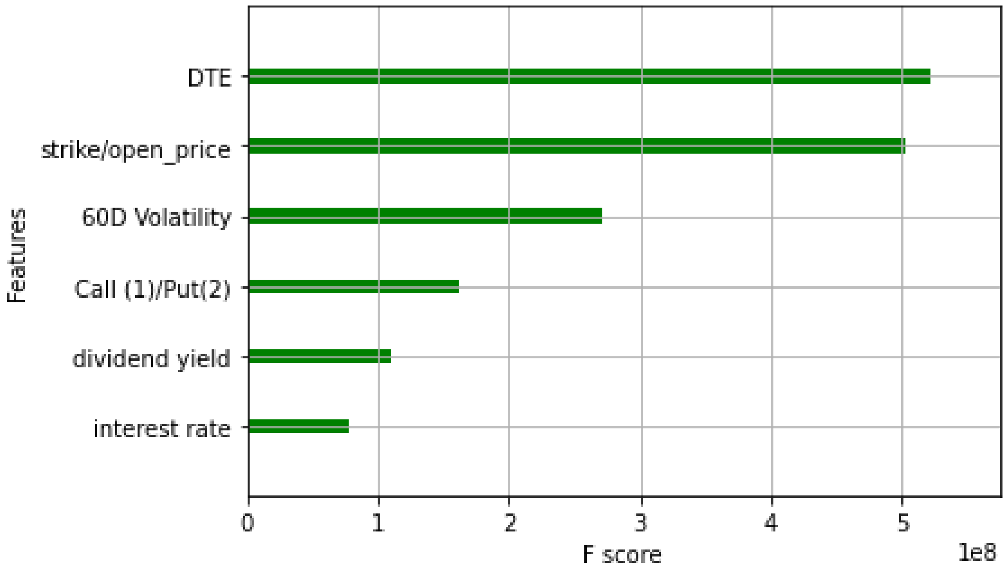


FIGURE 14: Feature importance graph by total coverage for the XGBoost Model 1

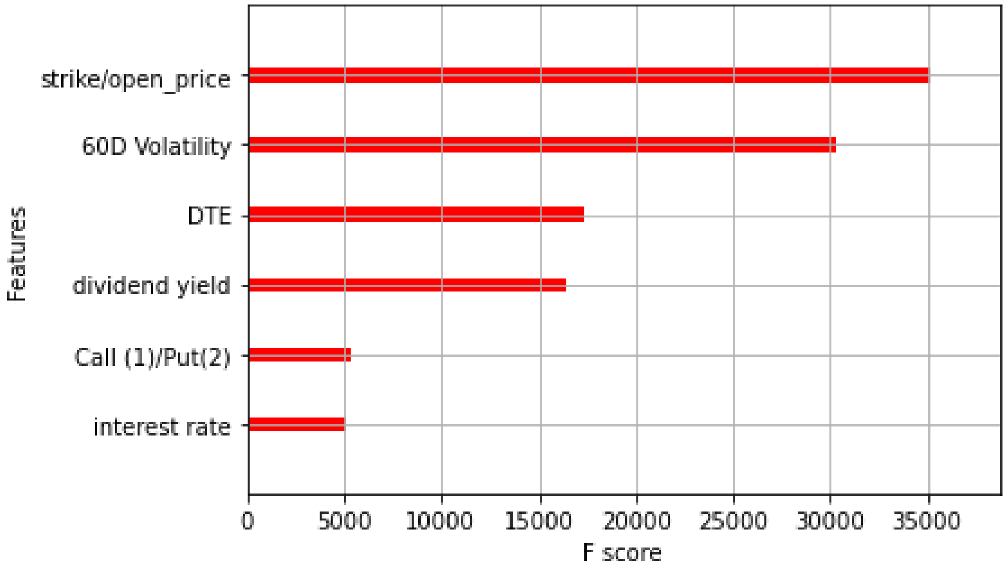


FIGURE 15: Feature importance graph by weight for the XGBoost Model 1

The model second XGBoost model has more variables than the previous model, but the most relevant variables by gain are the same: the distance to the strike or the level of moneyness, the Call/Put variable and the days to expiration, according to Figure 16. Although, the 90 day volatility is more relevant than the 60 day volatility, used in the first model. The two new variables related to the dividend: next dividend per share and the days until the ex-date are the least relevant by gain.

As in the first model, the dummy variable is quite relevant in the feature importance plots by coverage and gain, as shown in Figure 16 and 17, but it has a similar importance as the variables related to the stock price distribution. As in Figure 16, the days to expiration and the level of moneyness are extremely important to split the majority of the samples, according to Figure 17. Amongst the variables related to the distribution of the stock price but with low gain, the 90 day kurtosis and the 60 day volatility help split a lot of samples and are used very often, while the 30 day skewness is used very few times but helps splits almost as much samples as the other 2 variables.

Amongst the new variables related to the dividend, the days until the ex-date is the least important but also the least important among all variables. The days until the ex-date is the variable that provides the lowest gain, splits very few samples and is rarely used in the trees, so it is probably used at the bottom of the trees. The next dividend per share is a variable that helps split the samples almost as much as the variables related to the distribution of the stock price but is at the bottom, according to Figure 17. Although, it is used quite often in relation to the days until the ex-date because it's not at the bottom in the Figure 18.

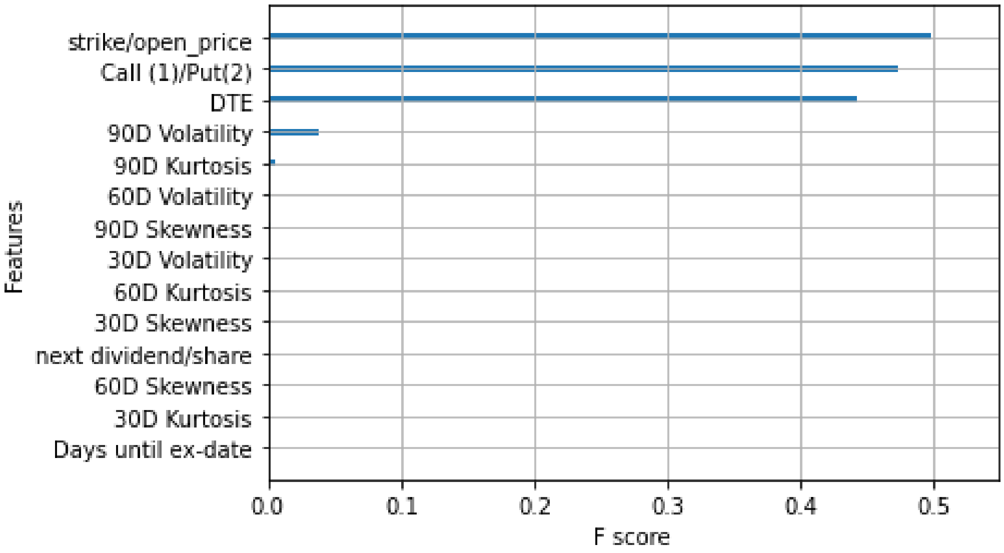


FIGURE 16: Feature importance graph by total gain for the XGBoost Model 2

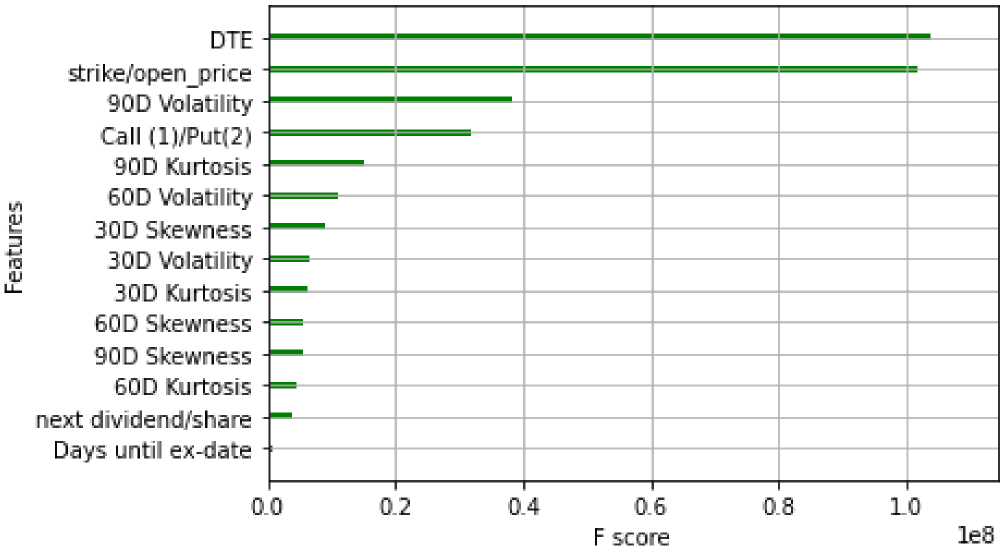


FIGURE 17: Feature importance graph by total coverage for the XGBoost Model 2

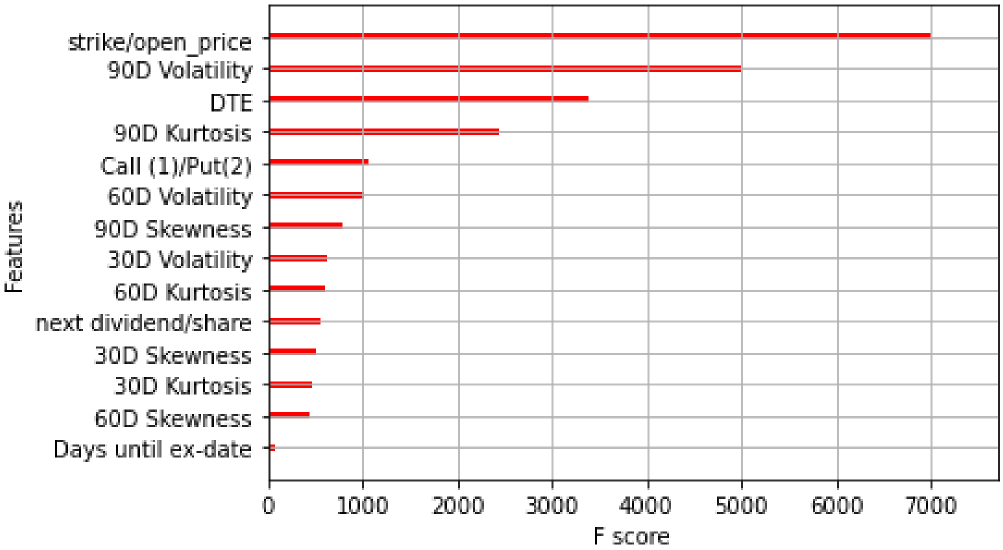


FIGURE 18: Feature importance graph by weight for the XGBoost Model 2

6 CONCLUSION

In this study it was shown how the XGBoost model could be used to forecast the price of vanilla options, using the Black-Scholes model as a benchmark. An explanation of the evolution of the XGBoost model and the Black-Scholes model was given. While the XGBoost model is a complex model that uses multiple features from previous ML models, the Black-Scholes model is simple and the first model of its kind.

In the discussion of results it was studied how accurate the models are using the box and whisker plots and how to understand the impact of each variable in the

XGBoost models using feature importance plots. Overall, the second XGBoost model performs better than the first model since it uses more inputs, but the BSM has the worst accuracy. The second XGBoost model has an error 8.6% and 8.5% lower than the simple XGBoost model based on the MAE and RMSE, respectively. In spite of using the same variables, the first XGBoost model is better by 29.3% and 29.5% than the BSM model based on the MAE and RMSE, respectively.

Both models fix a known weakness of the BSM related to the underpricing of OTM options. Although, both XGBoost models can't provide an accurate forecast with low standard-deviation for ITM options, despite being better than the BSM. For options with days to expiration higher than 102, XGBoost models are more accurate and have lower bias, but if lower than 8 days the bias is worse and the accuracy similar. Regardless of the size of the company, the bias and the accuracy of the models are the same, but with XGBoost models outperforming BSM.

In the last section of the discussion of results, the feature importance plots based on total gain are used to show that the level of moneyness, the days to expiration and the binary variable for calls and puts are the most important variables for both XGBoost models. The extra variables of the second model improve the option price forecasted by the first model and each one has a small gain but is used very often and affects a large number of samples.

References

Bakshi, Gurdip, Charles Cao, and Zhiwu Chen (1997). Empirical performance of alternative option pricing models. *The Journal of Finance* **52**(5), 2003–2049.

Black, Fischer and Myron Scholes (1973). The pricing of options and corporate liabilities. *Journal of Political Economy* **81**(3), 637–654.

Breiman, Leo (2001). Random forests. *Machine Learning* **45**(1), 5–32.

Chen, Tianqi and Carlos Guestrin (2016). Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp.785–794.

Chen, Tianqi and Carlos Guestrin (2021). *XGBoost Parameters - xgboost 1.6.1 documentation*. `https://xgboost.readthedocs.io/en/stable/parameter.html`.

Corrado, Charles J. and Tie Su (1996). Skewness and kurtosis in S&P 500 index returns implied by option price. *Journal of Financial Research* **19**(2), 175–192.

Culkin, Robert and Sanjiv R. Das (2017). Machine learning in finance: the case of deep learning for option pricing. *Journal of Investment Management* **15**(4), 92–100.

Derman, Emanuel and Iraj Kani (1994). Riding on a smile. *Risk* **7**(2), 32–39.

Freund, Yoav and Robert E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System sciences* **55**(1), 119–139.

Friedman, Jerome H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189–1232.

Gaspar, Raquel M, Sara D Lopes, and Bernardo Sequeira (2020). Neural network pricing of american put options. *Risks* **8**(3), 73.

Geske, Robert and Richard Roll (1984). On valuing american call options with the Black-Scholes European formula. *The Journal of Finance* **39**(2), 443–455.

Ho, Tin Kam (1995). Random decision forests. In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE, pp.278–282.

Hull, John and Alan White (1987). The pricing of options on assets with stochastic volatilities. *The Journal of Finance* **42**(2), 281–300.

Hull, John C. (2014). *Opções, futuros e outros derivativos*. Bookman Editora.

Ivașcu, Codruț-Florin (2021). Option pricing using machine learning. *Expert Systems with Applications* **163**, 113799.

Jarrow, Robert and Andrew Rudd (1982). Approximate option valuation for arbitrary stochastic processes. *Journal of Financial Economics* **10**(3), 347–369.

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* **30**.

Merton, Robert C. (1973). Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science* **4**(1), 141–183.

Naik, Vasanttilak (1993). Option valuation and hedging strategies with jumps in the volatility of asset returns. *The Journal of Finance* **48**(5), 1969–1984.

Park, Hyejin, Namhyoung Kim, and Jaewook Lee (2014). Parametric models and non-parametric machine learning models for predicting option prices: Empirical comparison study over kospi index options. *Expert Systems With Applications* **41**(11), 5227–5237.

Rätsch, Gunnar, Takashi Onoda, and Müller (2001). Soft margins for AdaBoost. *Machine Learning* **42**(3), 287–320.